

Параллельные вычислительные процессы

1

**РОМАНЕНКО ВЛАДИМИР
ВАСИЛЬЕВИЧ,
К.Т.Н., ДОЦЕНТ КАФ. АСУ ТУСУР**

Параллельные вычислительные процессы

2

ОПИСАНИЕ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ

Базовые определения

3

- Имена процессов будем обозначать словами, составленными из прописных букв, а буквами P, Q, R, \dots будем обозначать произвольные процессы.
- Буквы x, y, z, \dots используются для переменных, обозначающих события.
- Буквы A, B, C, \dots используются для обозначения множества событий.
- Буквы X, Y, Z, \dots используются для переменных, обозначающих процессы.
- Алфавит процесса P обозначается αP .
- Процесс с алфавитом αP , такой, что в нем не происходит ни одно событие из αP , назовем $СТОП_{\alpha P}$.

Базовые определения

4

Пример: автомат, торгующий шоколадками.

- В качестве имени процесса выберем *ТАП* (торговый аппарат простой).
- Имена событий – *мон* (опускание монеты в щель автомата) и *шок* (появление шоколадки из выдающего устройства).
- Алфавит $\alpha_{ТАП} = \{\text{мон}, \text{шок}\}$.

Префиксная запись

5

Префиксная форма описания процессов:

$$(x \rightarrow P),$$

где

● x – событие;

● P – процесс.

При этом

$$\alpha(x \rightarrow P) = \alpha P, x \in \alpha P.$$

Пример:

$$(мон \rightarrow (шок \rightarrow (мон \rightarrow (шок \rightarrow СТОП_{\alpha ТАП}))))).$$

Рекурсивная запись

6

Рекурсивный метод определения процесса:

$$P = (x \rightarrow P),$$

$$P = (x \rightarrow (y \rightarrow P)),$$

и т.д. Здесь $x, y \in \alpha P$.

Пример 1:

$$ТАП = (\text{мон} \rightarrow (\text{шок} \rightarrow ТАП)).$$

Рекурсивная запись

7

Рекурсивный метод определения процесса:

$$P = (x \rightarrow P),$$

$$P = (x \rightarrow (y \rightarrow P)),$$

и т.д. Здесь $x, y \in \alpha P$.

Пример 2: процесс **ЧАСЫ** описывает часы, единственная функция которых – тикать:

$$\alpha \text{ЧАСЫ} = \{\text{тик}\},$$

$$\text{ЧАСЫ} = (\text{тик} \rightarrow \text{ЧАСЫ}).$$

Определение выбора

8

Описание объектов с несколькими линиями поведения:

$$(x \rightarrow P \mid y \rightarrow Q),$$

где

$$\alpha(x \rightarrow P \mid y \rightarrow Q) = \alpha P, x, y \in \alpha P, \alpha P = \alpha Q.$$

Пример 2: копирование битов из входного канала в выходной:

$$\alpha \text{КОПИБИТ} = \{\text{вв.0}, \text{вв.1}, \text{выв.0}, \text{выв.1}\},$$

$$\text{КОПИБИТ} = ((\text{вв.0} \rightarrow \text{выв.0} \mid \text{вв.1} \rightarrow \text{выв.1}) \rightarrow \text{КОПИБИТ}).$$

Параллельные процессы

9

Оператор параллельной композиции:

$$P \parallel Q.$$

Законы:

- $P \parallel Q = Q \parallel P$ – логическая симметрия между процессом и его окружением;
- $(c \rightarrow P) \parallel (c \rightarrow Q) = c \rightarrow (P \parallel Q)$,
 $(c \rightarrow P) \parallel (d \rightarrow Q) = \text{СТОП}$ – пара процессов с одинаковыми алфавитами либо одновременно выполняет одно и то же действие, либо попадает в состояние тупика, если начальные события процессов не совпадают;

Параллельные процессы

10

Оператор параллельной композиции:

$$P \parallel Q.$$

Законы:

- $P \parallel (Q \parallel R) = (P \parallel Q) \parallel R$ – ассоциативность (при совместной работе процессов неважно, в каком порядке они объединены оператором параллельной композиции);
- $P \parallel \text{СТОП}_{\alpha P} = \text{СТОП}_{\alpha P}$ – процесс, находящийся в тупиковой ситуации, приводит к тупику всей системы.

Задача об обедающих философх

11

Алфавит философа:

$\alpha\text{ФИЛ}_i = \{i.\text{садится}, i.\text{встает}, i.\text{берет_вил.}i, i.\text{берет_вил.}(i+_{5}1),$
 $i.\text{кладет_вил.}i, i.\text{кладет_вил.}(i+_{5}1)\}$

Алфавит вилки:

$\alpha\text{ВИЛ}_i = \{i.\text{берет_вил.}i, (i-_{5}1).\text{берет_вил.}i,$
 $i.\text{кладет_вил.}i, (i-_{5}1).\text{кладет_вил.}i\}$

Задача об обедающих философях

12

Поведение философа:

$$\begin{aligned} \text{ФИЛ}_i = & (i.\text{садится} \rightarrow i.\text{берет_вил.}i \rightarrow i. \\ & \text{берет_вил.}(i+51) \rightarrow \\ & i.\text{кладет_вил.}i \rightarrow i.\text{кладет_вил.}(i+51) \rightarrow i. \\ & \text{встает} \rightarrow \text{ФИЛ}_i) \end{aligned}$$

Поведение вилки:

$$\begin{aligned} \text{ВИЛ}_i = & (i.\text{берет_вил.}i \rightarrow i.\text{кладет_вил.}i \rightarrow \text{ВИЛ}_i \mid \\ & (i-51).\text{берет_вил.}i \rightarrow (i-51).\text{кладет_вил.}i \rightarrow \\ & \text{ВИЛ}_i) \end{aligned}$$

Задача об обедающих философх

13

Поведение всего пансиона:

ФИЛОСОФЫ =
(ФИЛ₀ || ФИЛ₁ || ФИЛ₂ || ФИЛ₃ || ФИЛ₄),

ВИЛКИ =
(ВИЛ₀ || ВИЛ₁ || ВИЛ₂ || ВИЛ₃ || ВИЛ₄),

ПАНСИОН = (ФИЛОСОФЫ || ВИЛКИ)

Протоколы поведения процессов

14

Протоколом поведения процесса называется конечная последовательность символов, фиксирующая события, в которых процесс участвовал до некоторого момента времени.

Примеры:

- $\langle \rangle$ – пустой протокол;
- $\langle x \rangle$ – протокол из одного события;
- $\langle x, y \rangle$ – протокол из двух событий и т.д.
- $\langle \text{мон}, \text{шок}, \text{мон} \rangle$, $\langle \text{мон}, \text{шок}, \text{мон}, \text{шок} \rangle$.

Операции с протоколами

15

Конкатенация:

$$\begin{aligned}s \wedge t, \\ t^{n+1} &= t \wedge t^n, \\ (s \wedge t)^{n+1} &= s \wedge (s \wedge t)^n \wedge t.\end{aligned}$$

Например:

$$\langle \text{мон}, \text{шок} \rangle \wedge \langle \text{мон} \rangle = \langle \text{мон}, \text{шок}, \text{мон} \rangle.$$

Свойства:

- $s \wedge (t \wedge u) = (s \wedge t) \wedge u$ – ассоциативность;
- $s \wedge \langle \rangle = \langle \rangle \wedge s = s$ – пустой протокол служит единицей.

Операции с протоколами

16

Сужение:

$$t \uparrow A.$$

Например:

$$\langle \text{мон}, \text{шок}, \text{мон} \rangle \uparrow \langle \text{мон} \rangle = \langle \text{мон}, \text{мон} \rangle.$$

Свойства:

- $\langle x \rangle \uparrow A = \langle x \rangle$, если $x \in A$, $\langle y \rangle \uparrow A = \langle \rangle$, если $y \notin A$;
- $(s \wedge t) \uparrow A = (s \uparrow A) \wedge (t \uparrow A)$ – дистрибутивность;
- $\langle \rangle \uparrow A = \langle \rangle$, $s \uparrow \emptyset = \langle \rangle$, $(s \uparrow A) \uparrow B = s \uparrow (A \cap B)$.

Операции с протоколами

17

Голова и хвост:

s_o – первый элемент;

s' – результат, полученный после его удаления,
т.е. $\langle x, y \rangle_o = x$, $\langle x, y \rangle' = \langle y \rangle$.

Например:

$$\begin{aligned}\langle \text{мон}, \text{шок}, \text{мон} \rangle_o &= \text{мон}, \\ \langle \text{мон}, \text{шок}, \text{мон} \rangle' &= \langle \text{шок}, \text{мон} \rangle.\end{aligned}$$

Параллельные вычислительные процессы

18

СЕТИ ПЕТРИ

Основные определения

19

Сеть Петри N является четверкой $N = (P, T, I, O)$, где:

- $P = \{p_1, p_2, \dots, p_n\}$ – конечное множество *позиций*, $n \geq 0$;
- $T = \{t_1, t_2, \dots, t_m\}$ – конечное множество *переходов*, $m \geq 0$;
- $I: T \rightarrow P^*$ – входная функция, сопоставляющая переходу мультимножество его входных позиций;
- $O: T \rightarrow P^*$ – выходная функция, сопоставляющая переходу мультимножество его выходных позиций.

Основные определения

20

Граф сети Петри обладает двумя типами узлов: *кружок*, представляющий позицию сети Петри; и *планка*, представляющая переход сети Петри.

Маркировка μ – функция отображения

$$\mu: P \rightarrow Nat,$$

$$\mu = \langle \mu(p_1), \mu(p_2), \dots, \mu(p_n) \rangle,$$

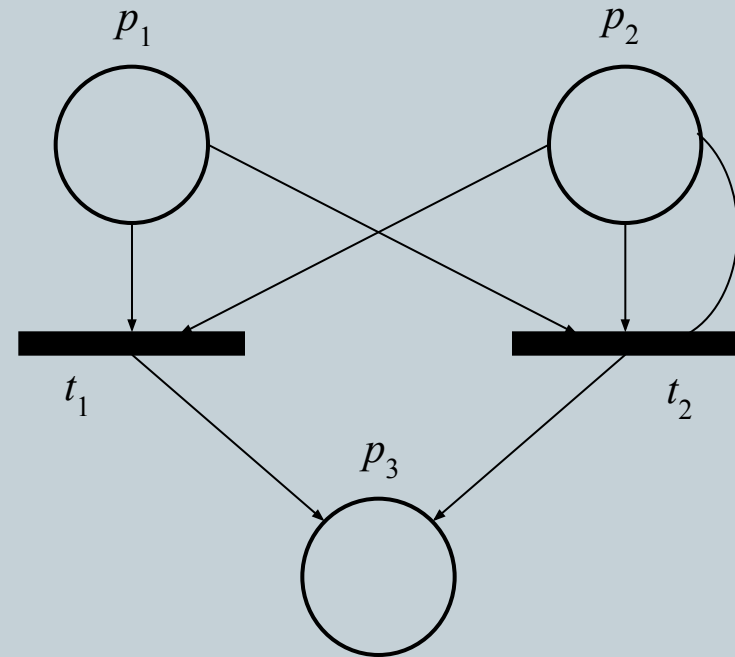
где n – число позиций в сети Петри и $\mu(p_i) \in Nat$,
 $1 \leq i \leq n$ – количество фишек в позиции p_i .

Основные определения

21

Пример: Сеть Петри $N = (P, T, I, O)$,

- $P = \{p_1, p_2, p_3\}$,
- $T = \{t_1, t_2\}$,
- $I(t_1) = \{p_1, p_1, p_2\}$,
 $O(t_1) = \{p_3\}$,
- $I(t_2) = \{p_1, p_2, p_2\}$,
 $O(t_2) = \{p_3\}$.

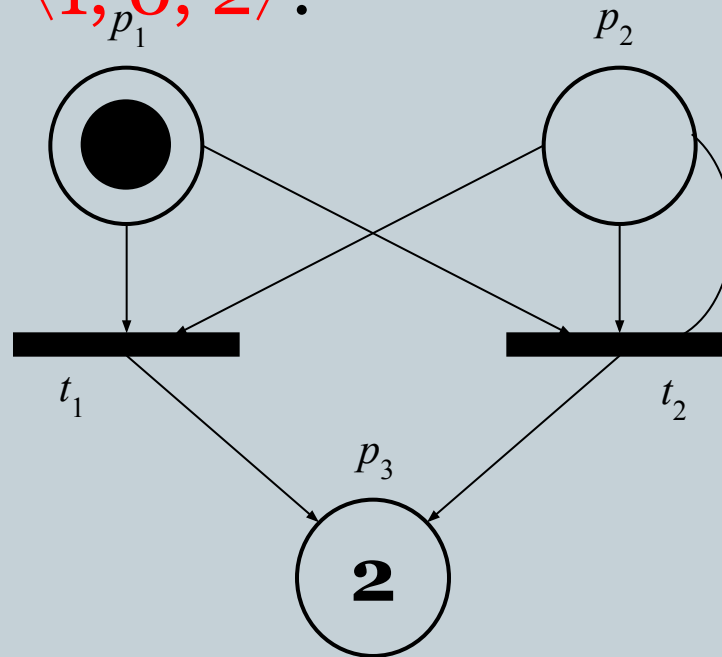


Основные определения

22

Маркированная сеть Петри $N = (P, T, I, O, \mu)$ определяется совокупностью структуры сети Петри $N = (P, T, I, O)$ и маркировки μ .

Например, $\mu = \langle 1, 0, 2 \rangle$:



Основные определения

23

Матричный вид сети Петри $N = (P, T, I, O)$ задается парой (D^-, D^+) , где:

- $D^-[k, i] = \#(p_i, t_k)$ – кратность дуги, ведущей из позиции p_i в переход t_k ;
- $D^+[k, i] = \#(t_k, p_i)$ – кратность дуги, ведущей из перехода t_k в позицию p_i ,

для произвольных $1 \leq k \leq m, 1 \leq i \leq n$. При этом

$$\mu' = \mu - e[k]D^- + e[k]D^+ = \mu + e[k]D.$$

Запуск сетей Петри

24

Сеть Петри *выполняется* посредством *запусков* переходов. При этом образуется новая маркировка μ' :

$$\mu'(p) = \mu(p) - \wedge \#(p, t) + \# \wedge(t, p),$$

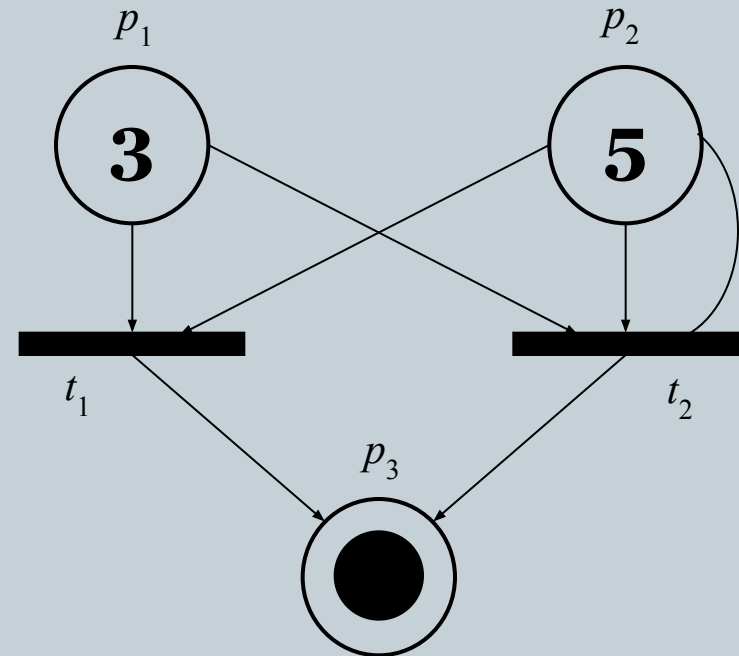
где:

- $\wedge \#: P \times T \rightarrow Nat$;
- $\# \wedge: T \times P \rightarrow Nat$;
- $\mu(p) \geq \wedge \#(p, t)$;
- $p \in P$.

Запуск сетей Петри

25

Пример: $\mu = \langle 3, 5, 1 \rangle$



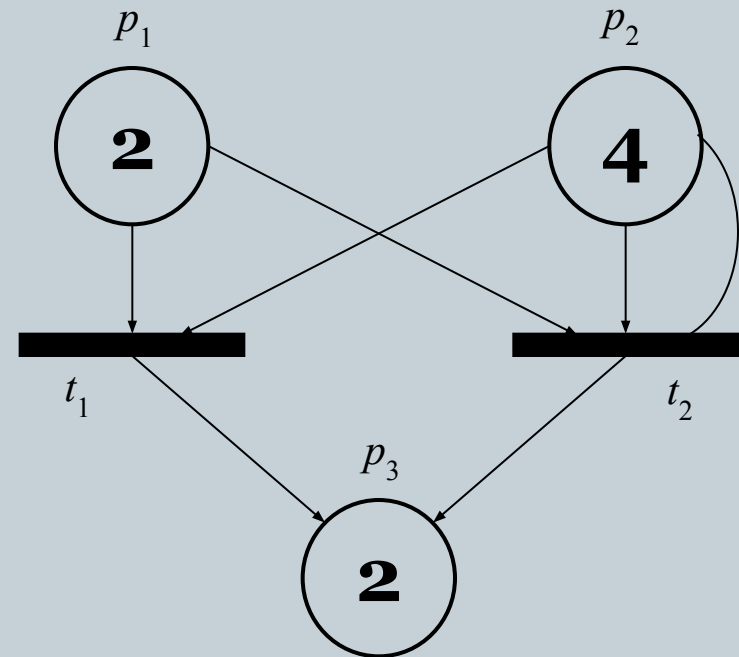
Запуск сетей Петри

26

Пример: $\mu = \langle 3, 5, 1 \rangle$

После перехода t_1 :

$\mu' = \langle 2, 4, 2 \rangle$



Запуск сетей Петри

27

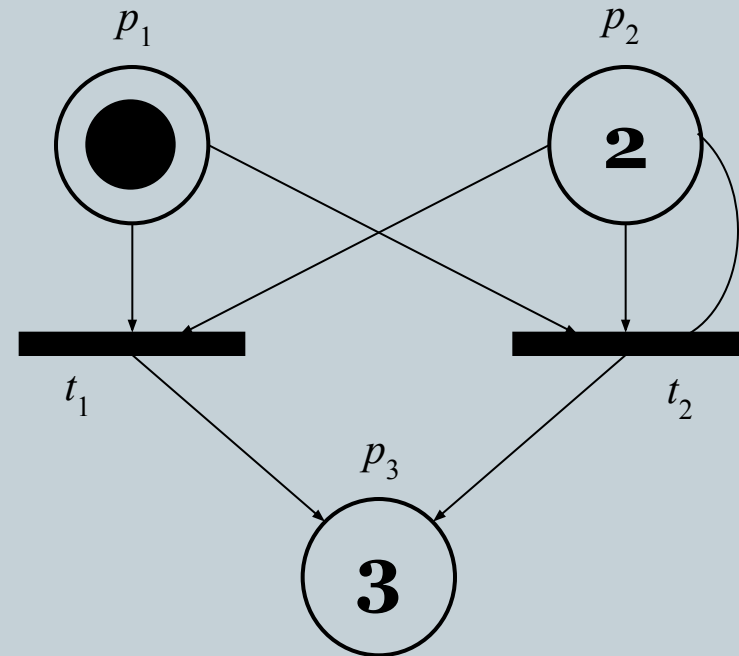
Пример: $\mu = \langle 3, 5, 1 \rangle$

После перехода t_1 :

$\mu' = \langle 2, 4, 2 \rangle$

После перехода t_2 :

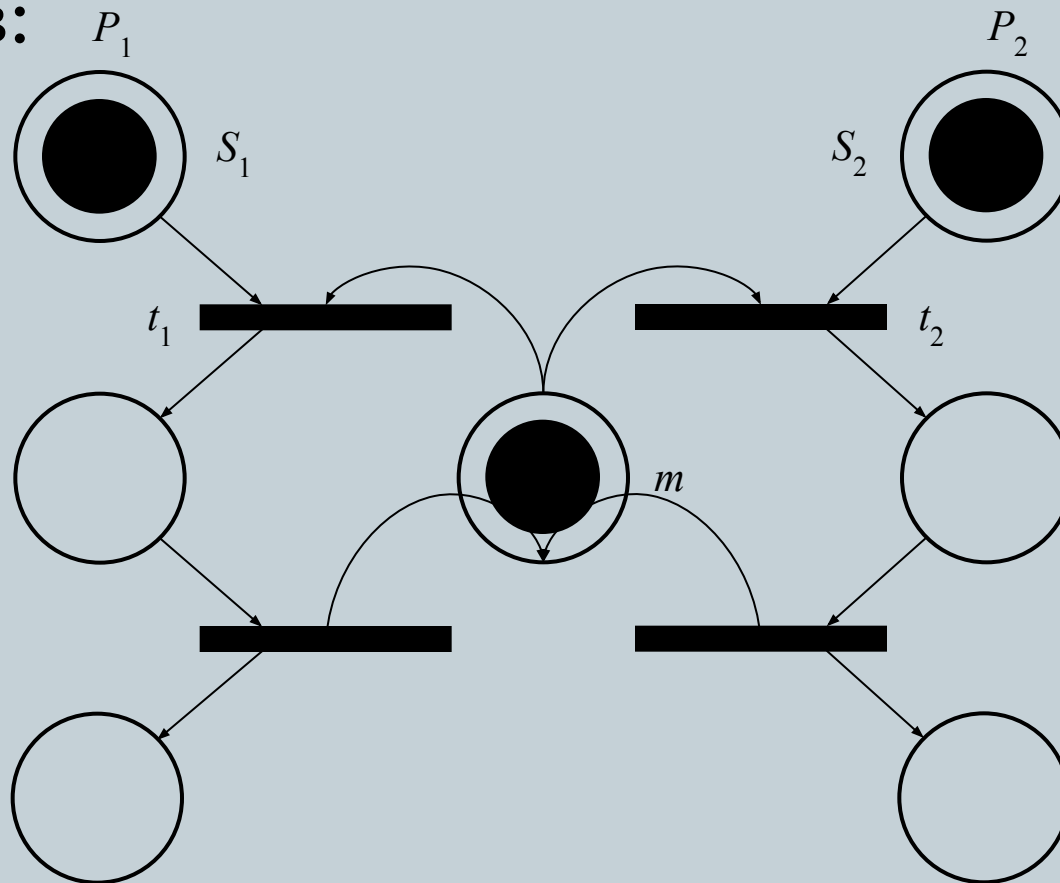
$\mu'' = \langle 1, 2, 3 \rangle$



Моделирование систем

28

Механизм взаимного исключения для двух процессов:



Моделирование систем

29

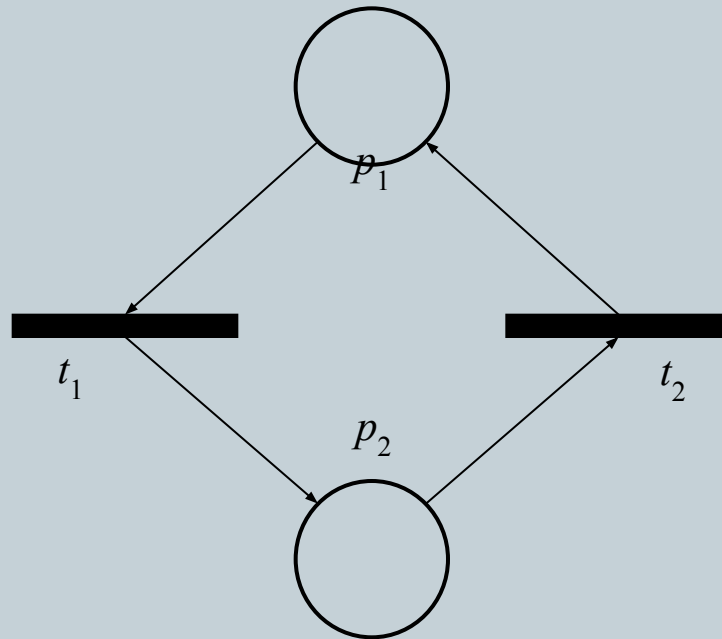
Простой семафор S :

- n – текущее значение счётчика;
- N_{max} – максимальное значение счётчика;
- P – операция блокирования семафора (WaitOne, WaitForSingleObject, acquire, ...);
- V – операция деблокирования семафора (Release, ReleaseSemaphore, release, ...).

Моделирование систем

30

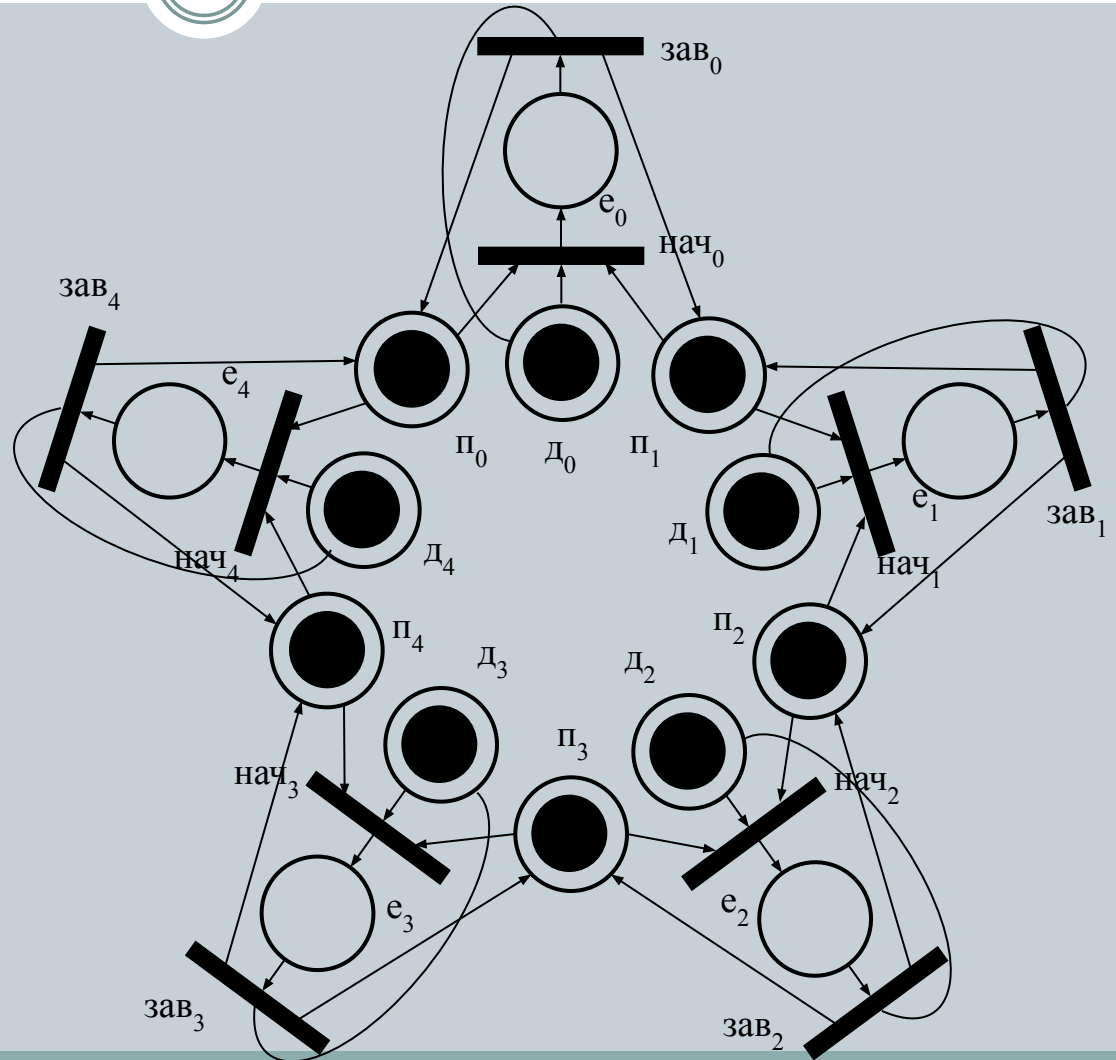
Простой семафор ***S***:



Моделирование систем

31

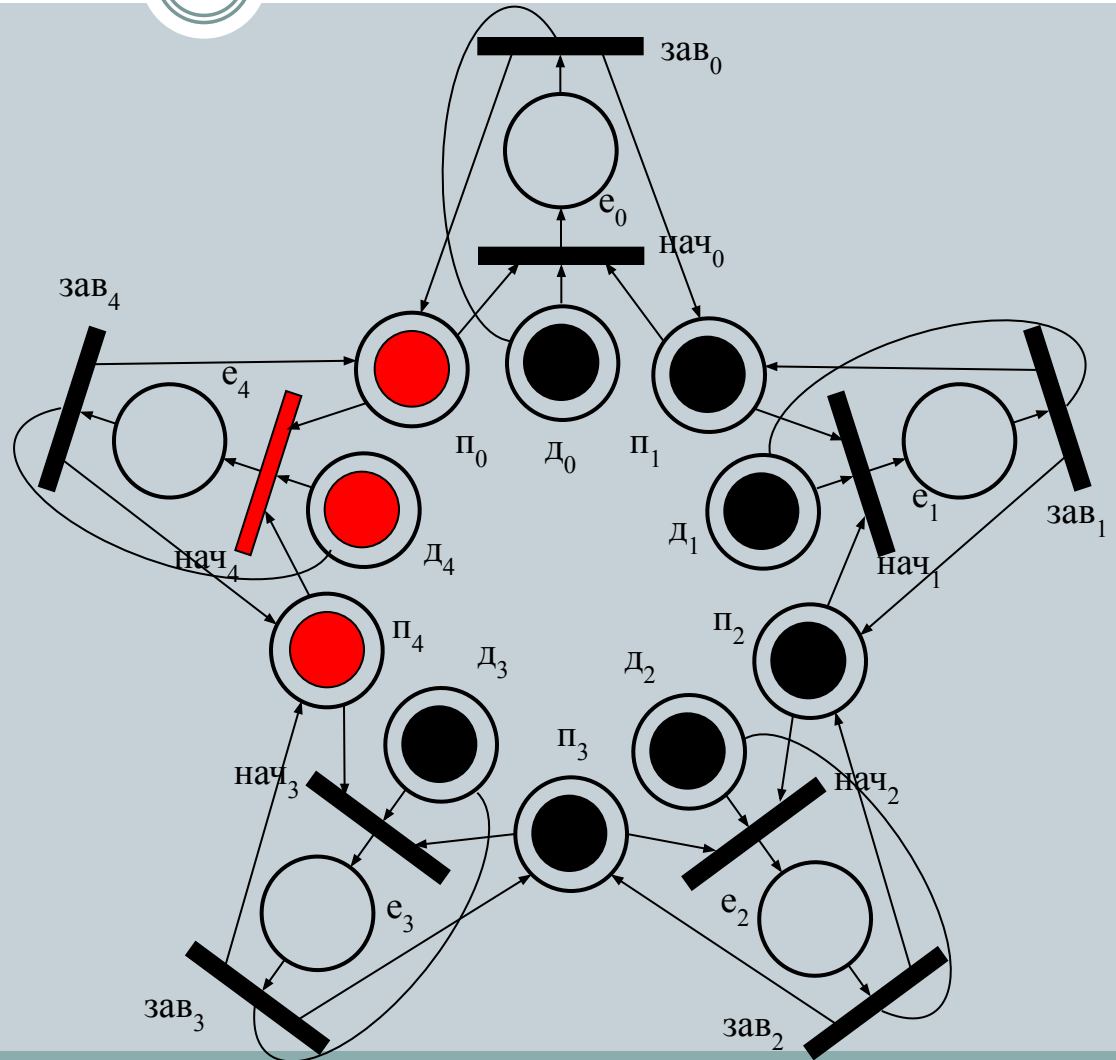
Задача об
обедающих
философах:



Моделирование систем

32

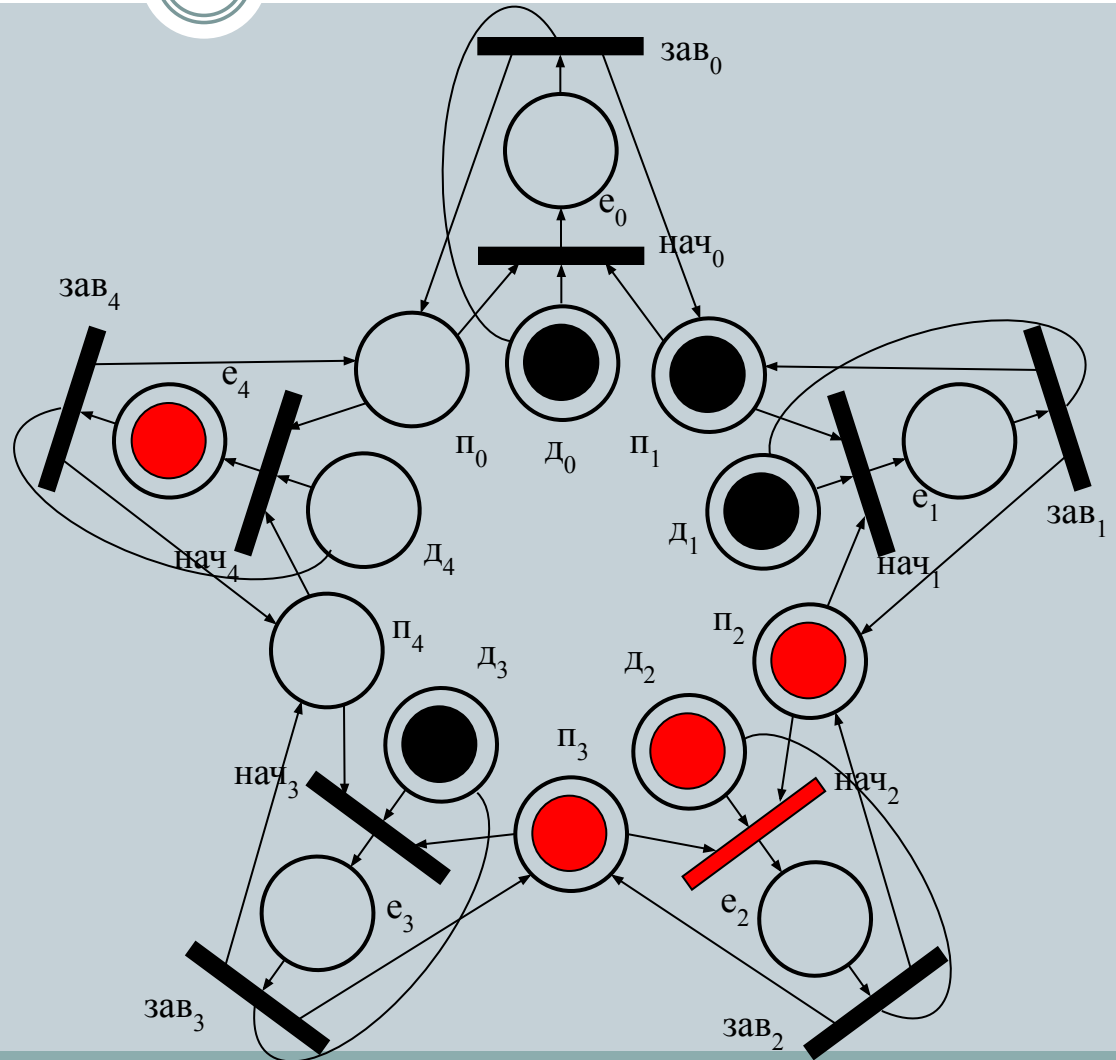
Задача об
обедающих
философах:



Моделирование систем

33

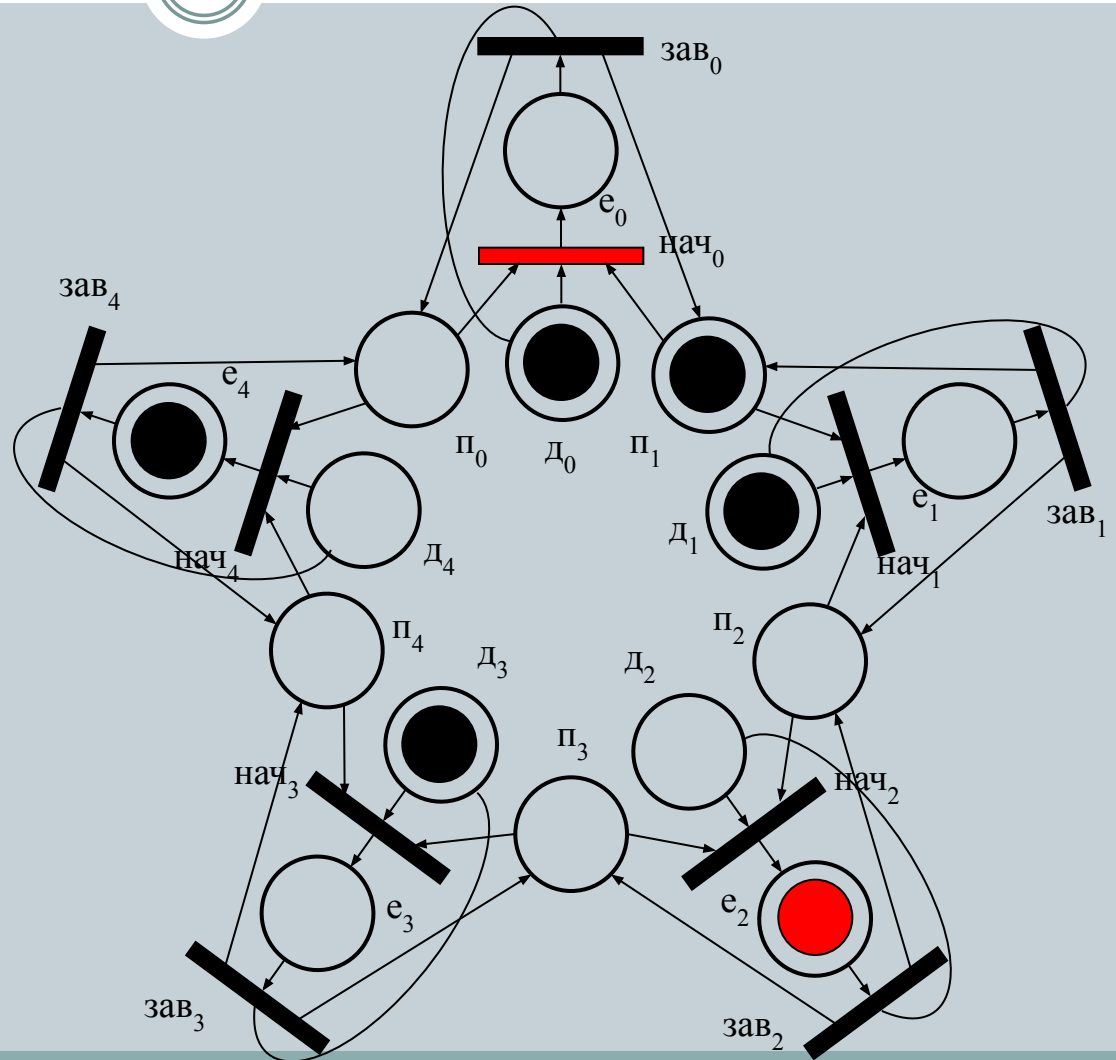
Задача об
обедающих
философах:



Моделирование систем

34

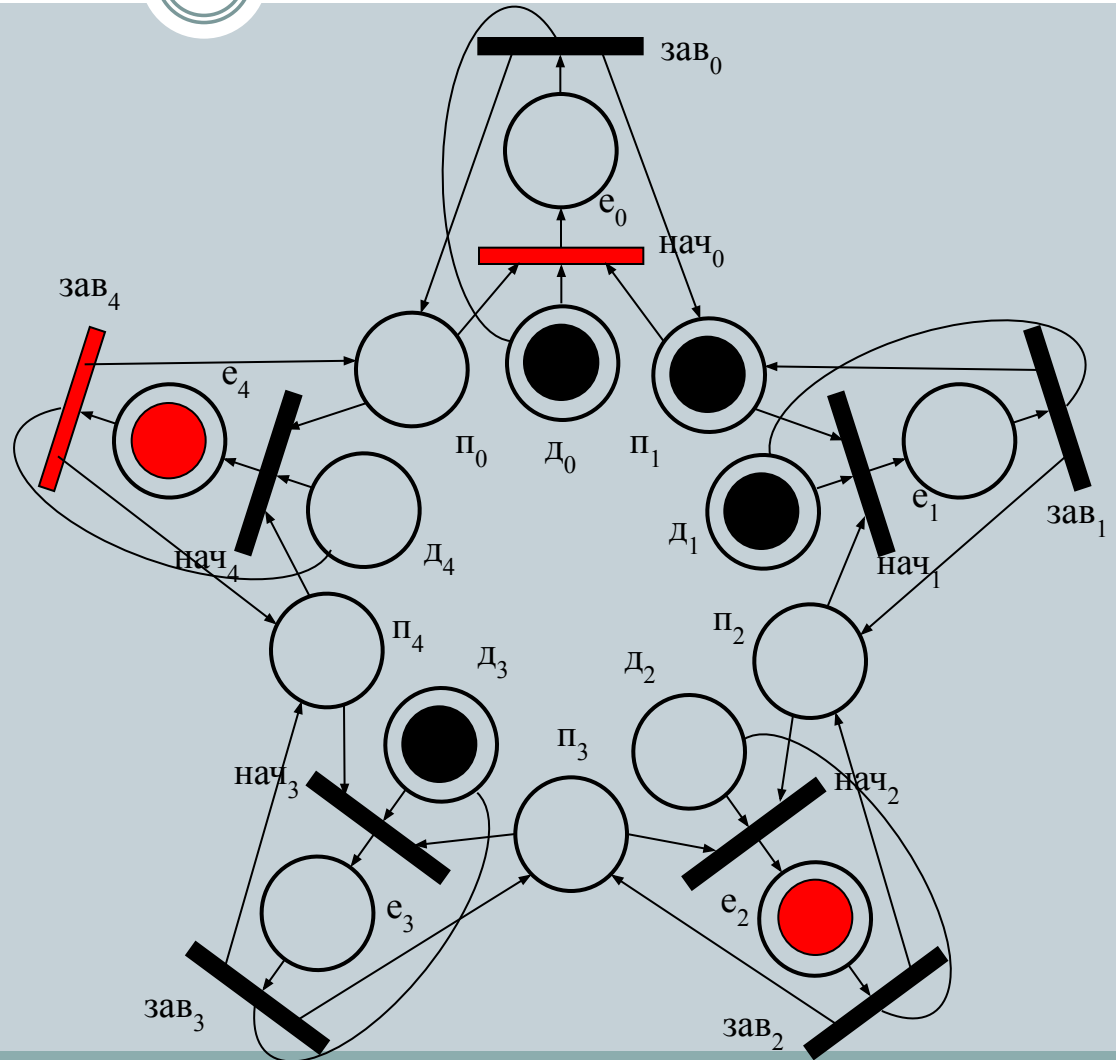
Задача об
обедающих
философах:



Моделирование систем

35

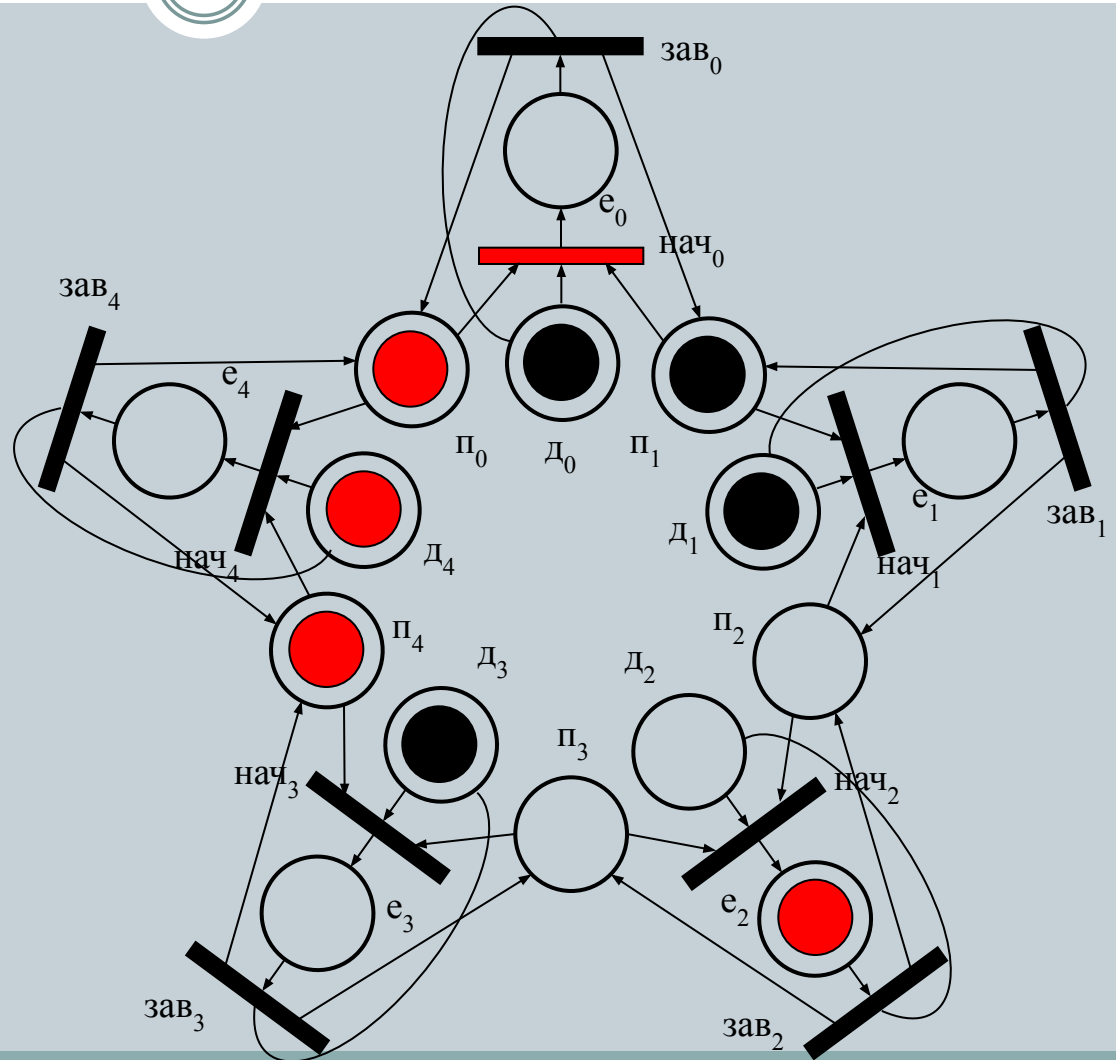
Задача об
обедающих
философах:



Моделирование систем

36

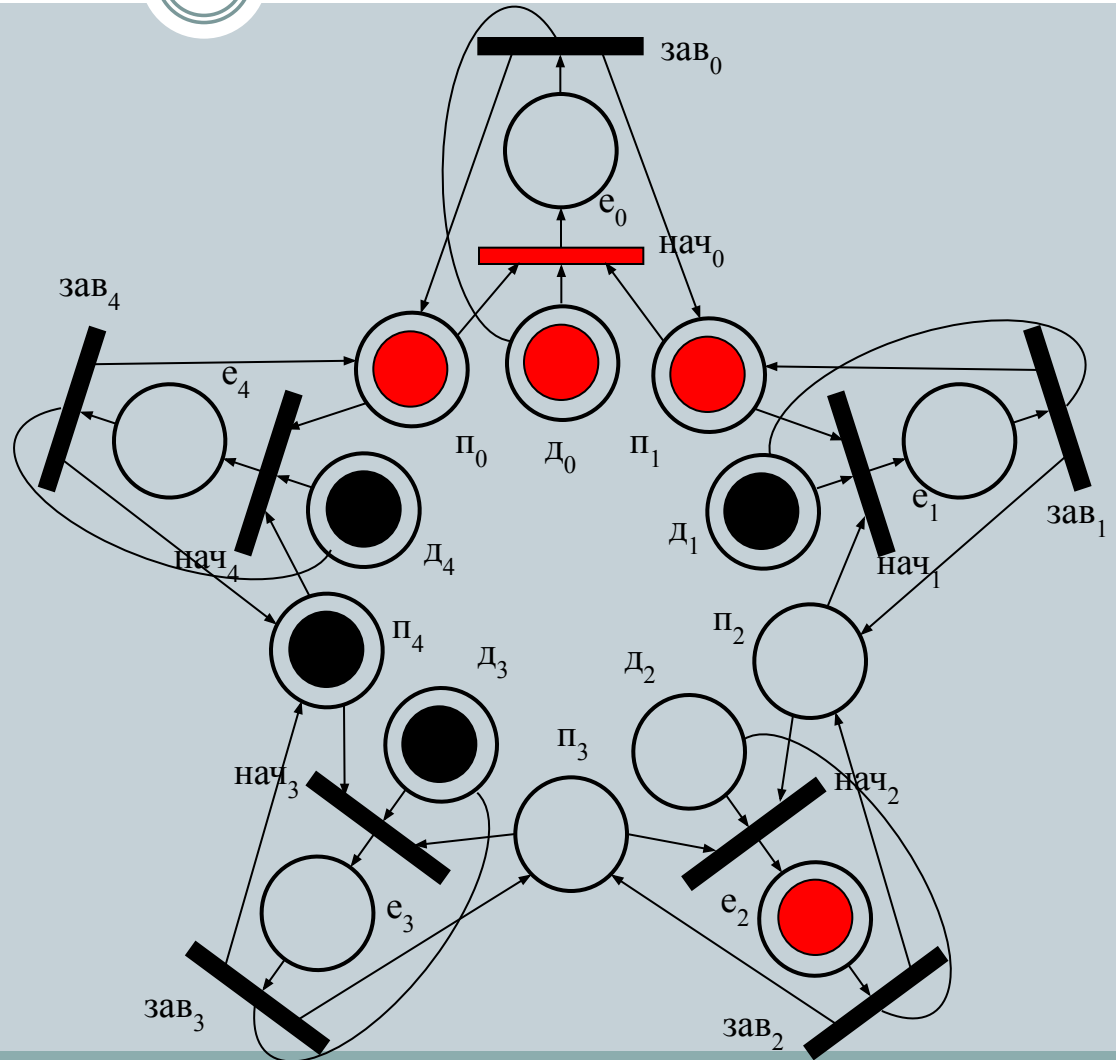
Задача об
обедающих
философах:



Моделирование систем

37

Задача об
обедающих
философах:



Моделирование систем

38

Задача об
обедающих
философах:

